

Synchronising Content Across Formats In-the-wild

Position Paper for «Programming» 2025's Software Substrates Workshop

Yann Trividic^a 

^a Université Paris-Saclay, CNRS, Inria, Laboratoire Interdisciplinaire des Sciences du Numérique

Abstract To collaborate while using multiple authoring tools often fails due to the inability to smoothly synchronise content across formats. In this position paper, I tackle a particular angle for substrates, namely *substrates for conversion*, as a means to support document editing workflows. I first outline major remaining challenges in format synchronisation, which suggest that we are still far from fully resolving these problems. I then identify more immediate interventions to be made to produce actionable results in the short term. Subsequently, I introduce the concept of *floss moves* as a way to apprehend individual-scale technical contributions to “substratify” existing artifact ecologies. This approach is grounded in my work on two projects I have been developing in the context of my Ph.D.: Propage and OutDesign, two systems that support the production of multi-format publications in the context of independent book publishing.

ACM CCS 2012

- Human-centered computing → User interface programming;
- Applied computing → Format and notation; Multi / mixed media creation;
- Software and its engineering → Interoperability;
- Information systems → Document structure;

Keywords publishing, software substrates, vision paper, format conversion

The Art, Science, and Engineering of Programming

Perspective The Art of Programming

Area of Submission Social Coding, General-purpose programming, User interface programming



© Yann Trividic
This work is licensed under a “CC BY 4.0” license
Submitted to *The Art, Science, and Engineering of Programming*.

1 Introduction

Contemporary digital work takes place in a world where heterogeneous software, operating systems, file formats, and practices coexist within an ever-changing technological stack. To collaborate effectively, people must navigate and compose through this diversity.

In the context of independent book publishing, which is my main field of interest, professionals collaborate by exchanging files across a range of formats, resulting in outputs that can take many different shapes [1]. Depending on their role in the book-making process, they work with formats such as PDF, INDD,¹ DOCX—respectively in order to copy edit, design the layout, or write the actual book—as no single format can encompass the full range of tasks that the publishing process requires [5]. Moreover, working with freelancers often implies not having complete control over the workflow, as independent professionals come with their own expertise, technical habits, and artifact ecologies [3, 14]. All of them work on content that at least partially overlap, but in contexts that are only partially compatible—one can import a DOCX file into an INDD file, and then export a PDF, but these are typically one-way processes among formats that are by nature significantly different.

These modalities of collaboration necessarily involve branching content, in order to merge them back once their peculiar tasks are done. In an ongoing study with our team, *ex)situ*, we argue that synchronising content across file formats is an overarching problem in knowledge work. At every stage of the collaboration process, loops emerge that were unforeseen by the linear workflows originally designed, and PDF serves in many cases as a fallback boundary object [15].

In this context, which has its own specificities but that is also in many ways extensible to other domains of knowledge work, I am interested in how software substrates can be conceived as facilitators to support more efficient content synchronisation, and thus, enable smoother collaboration.

2 On the horizon: Substrates for file format synchronisation

My approach to substrates is deeply influenced by the theoretical work by Mackay and Beaudouin-Lafon in particular [12]. Beaudouin-Lafon defines substrates as follows:

“A substrate is a digital computational medium that holds digital information, possibly created by another substrate, applies constraints and transformations to it, reacts to changes in both the information and the substrate, and generates information consumable by other substrates. Substrates are extensible, composable with other substrates, and they can be shared. They provide the fabric of the digital world.” [4]

In the context of content synchronisation, a substrate for file synchronisation would include (1) *constraints* that ensure valid schemas are maintained regardless of the

¹ Adobe InDesign native file format.

operation and (2) *transformations* that replicate operations applied in one format to the other. When working with text, or editorial content whatsoever, this addresses the problem of interoperability, and allows individuals to choose their preferred application for working on their document, without preventing collaboration.

For example, an author may choose to work in either a DOCX file, or a lightweight markup language such as Markdown. For a graphic designer, fulfilling their role might include IDML² or CSS files.

Ideally, such content could be synchronised flawlessly, supporting multiple modes of collaboration depending on the project's stage. These functionalities—when they even exist—currently require setting up a tightly controlled environment, in which file formats are sometimes imposed on the users. However, imposing an unfamiliar workflow on expert workers often leads to deskilling, and thus cannot constitute a default proposition [14].

In a perfect world, those transformations would systematically track data provenance and operate bidirectionally. Work has been done in this direction [6, 8, 11], but to my knowledge, most of these approaches rely on *ad hoc* environments, or require strong technical prerequisites to be met—therefore contradicting the need for flexible solutions—making them the vision of a long-term horizon.

3 At our doorstep: Substrates for file format conversion

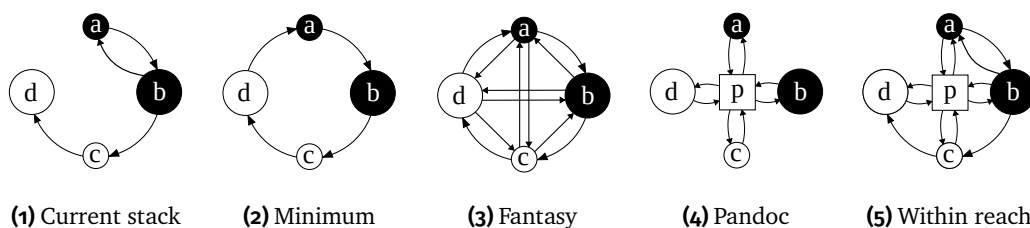
My focus is on a lower-hanging fruit: conversion within the existing artifact of ecology [3]—that is, making the current technological stack more interoperable. Converting *in-the-wild* implies to make some trade-offs, as fully fledged conversion is rarely achievable [11].

In Figure A, I sketch what I believe could be an interesting take at what constitutes a *substrate for conversion*. Conversion tasks (import, export, convert to) are typically seen as isolated, native features of specific tools. Yet the technological stack, taken as a whole, already offers rich possibilities to convert from and to a broad range of formats. By building upon the sum of all converters already available on our personal computers, most of the material is already present, *within reach* (Fig. A5), to bridge some gaps between authoring tools.

Building on existing technologies does require compromising for certain features, and can sometimes be less straightforward than more intuitive prototyping. However, the benefits of relying on existing codebases and functionalities are considerable, as also noted in Heer and al. [7]. In another paper, Heer and Horowitz refer to the idea of considering the stack in Engraff as the “*composition of tools and environments in the outside world*” [9]. By advocating for interoperability, we can help avoid deskilling, while giving individuals a gentler learning curve when adapting their workflow is necessary.

² IDML stands for InDesign Markup Language. It is Adobe InDesign's open format “*to facilitate the inspection and construction of InDesign content outside of InDesign*”. [2]

Synchronising Content Across Formats In-the-wild



■ **Figure A** Four file formats are represented in a series of graphs as four nodes (a), (b), (c), and (d). An extra [p] node represents the Pandoc AST. **Figure A1** illustrates the range of conversion possibilities usually available in a “casual” artifact ecology. Some formats are dead-ends, while some conversion paths form cycles. **Figure A2** outlines a dreamt world, in which one would only require a single Hamiltonian cycle, minimising the number of converters to the number of formats. Though, most conversion processes lose data in the process, and such a cycle would only preserve the intersection of information that is encoded by all formats. Also, users might want to prioritize some paths over others. **Figure A3** presents a complete fantasy graph that projects a complete ensemble of bidirectional conversions, and raises many questions currently without answers. **Figure A4** introduces Pandoc’s model, a many-to-many formats conversion engine that uses a pivot abstract syntax tree as a central node. **Figure A5** computes the union of A1 and A4.

Obviously, this angle comes with trade-offs, but presents the advantage of enabling direct, pragmatic, and actionable interventions. It does not aim to solve all problems—particularly those requiring fundamental changes to the stack—but it “substratifies” existing toolchains.

Thus, I introduce the concept of *floss moves*,³ code contributions that bring in-the-wild authoring tools closer to more malleable digital environments [10, 16]. Such moves require a clear view of both the state-of-the-art software stack and the practices of its users, in order to identify and isolate the minimal interventions to unlock or support practices and use cases—in the idea of performing a kind of “judo hold” to the stack [13].

4 Ongoing technical contributions

As part of this research, I developed two systems that target some specific issues in publishing work: Propage and OutDesign.

4.1 Propage

This first project addresses the need of an operational publishing workflow for the publishing house I am part of: éditions Burn~Août, an associative French publishing

³ FLOSS stands for *Free, Libre, and Open Source Software*.

house. Propage is a multiformat publishing framework that aims to support independent publishers wishing to disseminate their texts to their readership by automating the production of a selection of formats (for the web, for print, for re-use), and to cover all situations that may hinder access to a text. In other words, Propage is designed for small publishing structures wanting to facilitate access to their books in a way that is agnostic of their readers' access conditions, be they economic, physical, material or geographical. The system includes a visualiser that shows content and their provenance in different file formats, with a common source in AsciiDoc,⁴ and n arbitrary views. Each view scrolls synchronously with the others, regardless of its format—be it a regular webpage, paginated ready-for-print HTML, or a PDF.

To support this work, I contributed code to WeasyPrint,⁵ enabling the embedding of metadata in PDF objects to allow turning WeasyPrint into an interactive PDF generator.

4.2 OutDesign

My second project is a system that helps reading IDML files through Pandoc,⁶ in order to facilitate the reuse of InDesign documents, that might otherwise remain captive in Adobe's walled garden. OutDesign is a *get-out-of-jail-free card* for liberating editorial content, allowing it to be edited in any authoring tool of choice.

To support this work, I contributed code to Pandoc,⁷ to enhance semantic support for DocBook, which I use as an intermediary format to generate a Pandoc AST from IDML files.

I now plan to extend this work to an interface for schema mapping, in order to allow for conversion customisation.

5 Conclusion

In this paper, I have argued for an incremental approach to substrates—one that builds on the existing software ecosystem. By introducing the notion of *floss moves*, I have outlined how targeted interventions can “substratify” everyday authoring environments without imposing new constraints on individuals.

I look forward to continuing this discussion at the *Software Substrates* workshop, where I hope to share these thoughts and experiments, and learn from other perspectives on how to make our workflows more collaborative and coherent with real-world constraints.

⁴ <https://asciidoc.org>

⁵ <https://github.com/Kozea/WeasyPrint/pull/2192>

⁶ <https://pandoc.org>

⁷ <https://github.com/jgm/pandoc/pull/10665>

References

- [1] Janneke Adema. *Living Books: Experiments in the Posthumanities*. The MIT Press, Aug. 2021. ISBN: 978-0-262-36646-5. DOI: 10.7551/mitpress/11297.001.0001. (Visited on 2023-11-14).
- [2] Adobe. *InDesign Markup Language (IDML) Cookbook*. 2010. (Visited on 2025-02-16).
- [3] Liam J. Bannon and Susanne Bødker. “Beyond the Interface: Encountering Artifacts in Use”. In: *DAIMI Report Series* 288 (Oct. 1989). ISSN: 2245-9316. DOI: 10.7146/dpb.v18i288.6666. (Visited on 2023-10-13).
- [4] Michel Beaudouin-Lafon. “Towards Unified Principles of Interaction”. In: *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*. Cagliari Italy: ACM, Sept. 2017, pages 1–2. ISBN: 978-1-4503-5237-6. DOI: 10.1145/3125571.3125602. (Visited on 2023-04-14).
- [5] Douglas C. Engelbart. “Knowledge-Domain Interoperability and an Open Hyperdocument System”. In: *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work*. CSCW ’90. New York, NY, USA: Association for Computing Machinery, Sept. 1990, pages 143–156. ISBN: 978-0-89791-402-4. DOI: 10.1145/99332.99351. (Visited on 2024-12-20).
- [6] Julien Gori, Han L. Han, and Michel Beaudouin-Lafon. “FileWeaver: Flexible File Management with Automatic Dependency Tracking”. In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. Virtual Event USA: ACM, Oct. 2020, pages 22–34. ISBN: 978-1-4503-7514-6. DOI: 10.1145/3379337.3415830. (Visited on 2024-01-26).
- [7] Jeffrey Heer, Matthew Conlen, Vishal Devireddy, Tu Nguyen, and Joshua Horowitz. “Living Papers: A Language Toolkit for Augmented Scholarly Communication”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST ’23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pages 1–13. ISBN: 979-8-4007-0132-0. DOI: 10.1145/3586183.3606791. (Visited on 2023-11-14).
- [8] Brian Hempel, Justin Lubin, and Ravi Chugh. “Sketch-n-Sketch: Output-Directed Programming for SVG”. In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST ’19. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pages 281–292. ISBN: 978-1-4503-6816-2. DOI: 10.1145/3332165.3347925. (Visited on 2025-03-05).
- [9] Joshua Horowitz and Jeffrey Heer. “Engraft: An API for Live, Rich, and Composable Programming”. In: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. UIST ’23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pages 1–18. ISBN: 979-8-4007-0132-0. DOI: 10.1145/3586183.3606733. (Visited on 2025-03-21).

- [10] Stephen Kell. “Convivial Design Heuristics for Software Systems”. In: *Conference Companion of the 4th International Conference on Art, Science, and Engineering of Programming*. Porto Portugal: ACM, Mar. 2020, pages 144–148. ISBN: 978-1-4503-7507-8. DOI: 10.1145/3397537.3397543. (Visited on 2024-03-04).
- [11] Geoffrey Litt, Peter van Hardenberg, and Orion Henry. “Cambria: Schema Evolution in Distributed Systems with Edit Lenses”. In: *Proceedings of the 8th Workshop on Principles and Practice of Consistency for Distributed Data*. PaPoC ’21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pages 1–9. ISBN: 978-1-4503-8338-7. DOI: 10.1145/3447865.3457963. (Visited on 2025-02-24).
- [12] Wendy Mackay and Michel Beaudouin-Lafon. “Interaction Substrates: Combining Power and Simplicity in Interactive Systems”. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*. CHI ’25. ACM, To appear.
- [13] Nolwenn Maudet. “Design tactique”. In: *Tèque 3*. Volume 3. Tèque. Éditions Présentes, Jan. 2023, pages 40–69. ISBN: 97824924691833. (Visited on 2023-01-30).
- [14] Midas Nouwens and Clemens Nylandsted Klokmoose. “The Application and Its Consequences for Non-Standard Knowledge Work”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI ’18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pages 1–12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173973. (Visited on 2024-11-05).
- [15] Susan Leigh Star and James R. Griesemer. “Institutional Ecology, ‘Translations’ and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, 1907-39”. In: *Social Studies of Science* 19.3 (Aug. 1989), pages 387–420. ISSN: 0306-3127. DOI: 10.1177/030631289019003001. (Visited on 2024-02-19).
- [16] Philip Tchernavskij. “Designing and Programming Malleable Software”. PhD thesis. Université Paris Saclay (COMUE), Dec. 2019. (Visited on 2023-11-13).

Synchronising Content Across Formats In-the-wild

About the author

Yann Trividic Ph.D. candidate in the ex)situ team, LISN.

More info: <https://yanntrividic.fr>. Contact him at ytrividic@liscn.fr.

 <https://orcid.org/0000-0003-0110-6918>

***** PAPER 10 *****

AUTHORS: Yann Trividic

TITLE: Synchronising Content Across Formats In-the-wild

+++++++ REVIEW 1 (Gilad Bracha) +++++++

Same review as for #11 - my perspective on both papers is the same.

This (and the other essay from INRIA) provide a very different perspective on substrates. Here the emphasis on document production and the difficulties around varying formats. Essentially nothing is said about computation; the idea of integrating documents and computation is core to the idea of substrates. So in a way this almost irrelevant; but I see real value in getting the experience of people dealing with real problems.

My own take on this is very modernist: build tools that do all this properly rather than battling all this legacy. However, humans are already tied to the old broken ways. AI will make resolving this quite straightforward in the foreseeable future.

+++++++ REVIEW 3 (Tomas Petricek) +++++++

The vision statement argues that we need to find a way of turning existing software ecosystems into "substrate" by gradual adaptation - instead of thinking of a "new ultimate substrate to replace the entire stack". The statement then suggests "floss move" as a specific intervention to get there. In the specific case discussed in the statement (that is, compatibility of different formats), this may involve adding support for particular format conversion.

This is a very nice paper - and I really like the idea of "how to substratify the existing stack". The paper suggests some themes worth thinking about:

* Methodology - how can post-modern approaches (presented here) interact with modernist approaches (presented in some of the other submissions)? Related to this, I'm also curious how would the author classify Kell's work on UNIX - is it post-modern in that it embraces existing systems, or modernist in that it seeks to intervene at one point?

* The problem of format synchronization seems to be very close to our idea of "overlapping notations" from the "notational structure" technical dimension [1]. This also relates it to various other work on this problem in programming systems (such as systems that present programs as code and UML diagram at the same time). Those suffer from the problem of where to store the additional information that is needed in only one view. We may need some more general mechanism for this - substrates should be flexible enough to allow for storing extra information they cannot understand? (I'm vaguely reminded of [2].)

I think both the methodology suggested in the paper and the specific problem of format synchronization are great topics for discussion at the meeting.

[1] Jakubovic, Joel, Jonathan Edwards, and Tomas Petricek. "Technical Dimensions of Programming Systems." The Art, Science, and Engineering of Programming 7.3 (2023): 13-1.

[2] Hall, Christopher, Trevor Standley, and Tobias Hollerer. "Infra: Structure all the way down: Structured data as a visual programming language." Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. 2017.

+++++++ REVIEW 4 (Tom Larkworthy) +++++++

Substrate in this context seems to be a conduit of digital information. Social media seems to fit. But here we are talking about applied representational transforms for documents. The diagram explaining the possible topologies is excellent for framing the work. I don't understand the term "floss moves", I am thinking it might be invoking "Free/Libre Open Source Software" but I am not sure.

I think the work is interesting to substrate research because a network of converts is a "level" above the set of digital representations, and there a practical substrate arises. It hits technical challenges immediately. Every in-the-wild representation has specialization, and each individual bridge is lossy. I think comparisons to FFMPEG might be useful, which deals with a similar fragmented ecosystem.

+++++++ REVIEW 5 (Clemens Nylandsted Klokmoose) +++++++

This paper presents a pragmatic approach to try to "subtratify" an existing software stack for publishing by using Pandoc as a sort of lingua franca between existing platforms. I think this is an interesting, and probably right-minded, approach to tackling a difficult challenge of interoperability. I would love to see demos of Propage and OutDesign.

I didn't fully understand the concept of floss moves. It sounds compelling, but it needs to be developed further. A lot of tools for thought and related work takes Markdown as its basic text substrate. I think its nice to see AsciiDoc explored as an alternative.